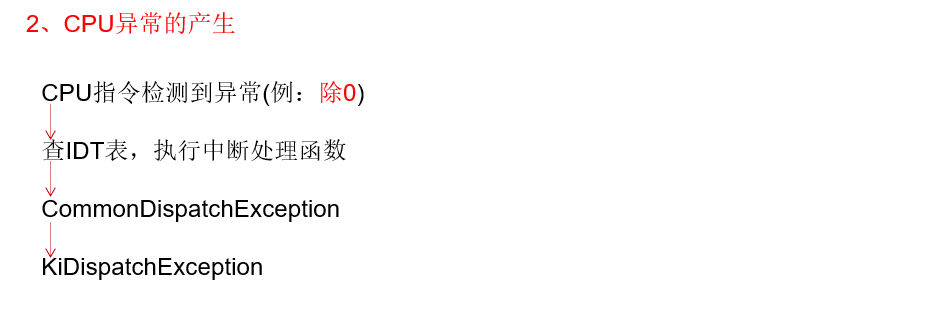
例题：

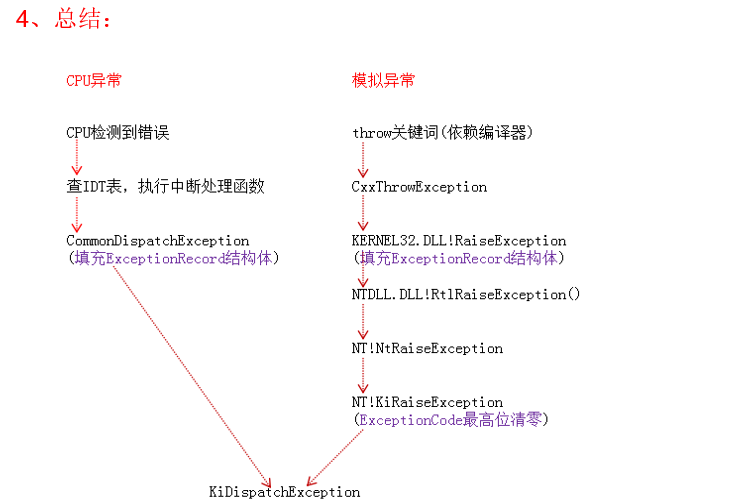
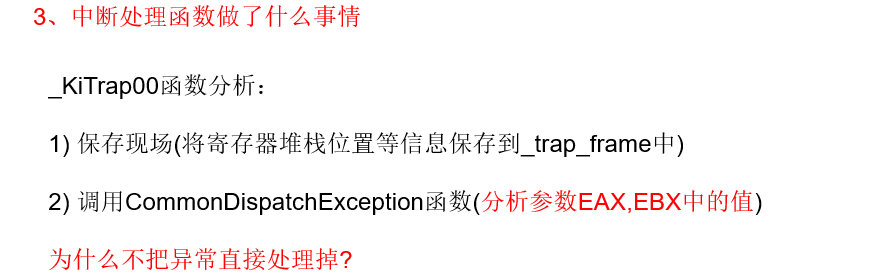
1. 新春赛的
2. 安洵杯的



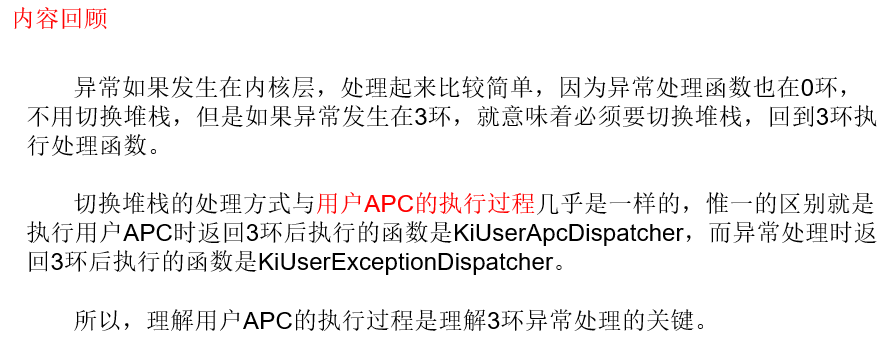




中断程序的作用;

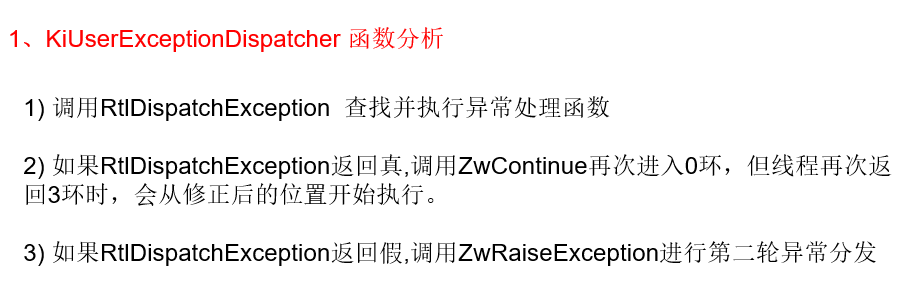


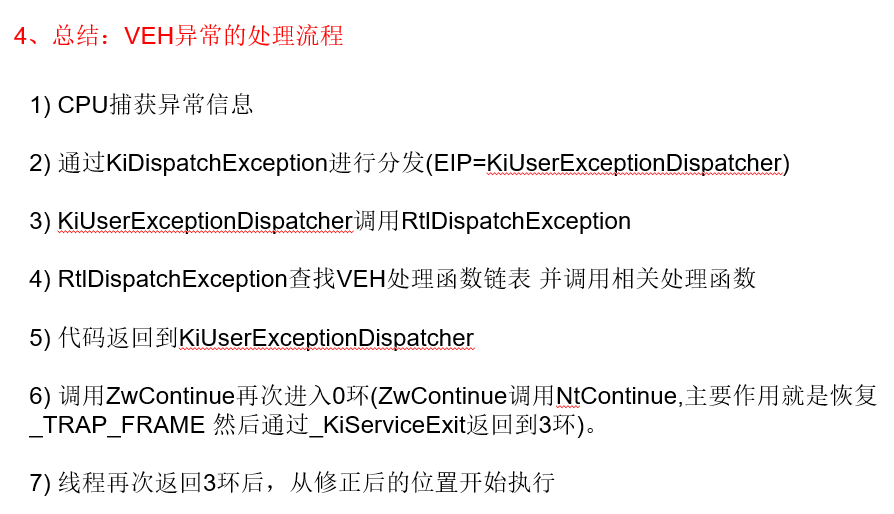


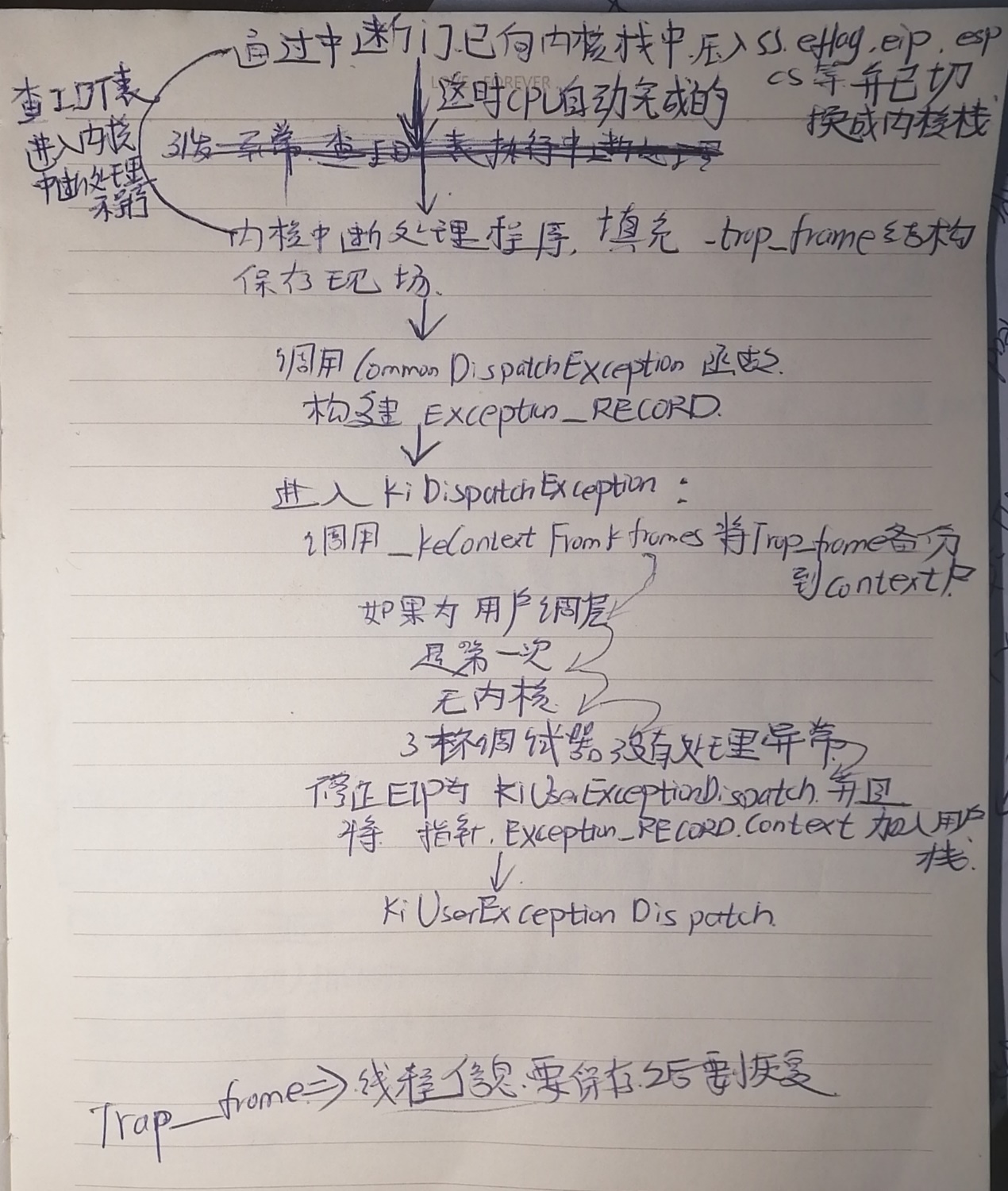








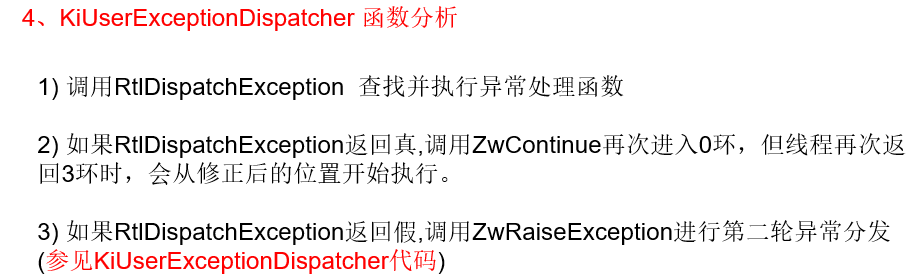


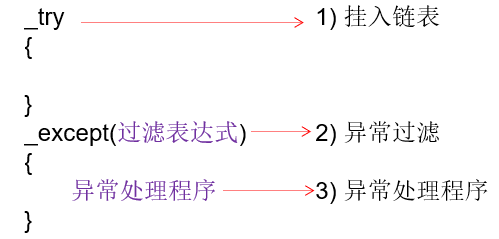


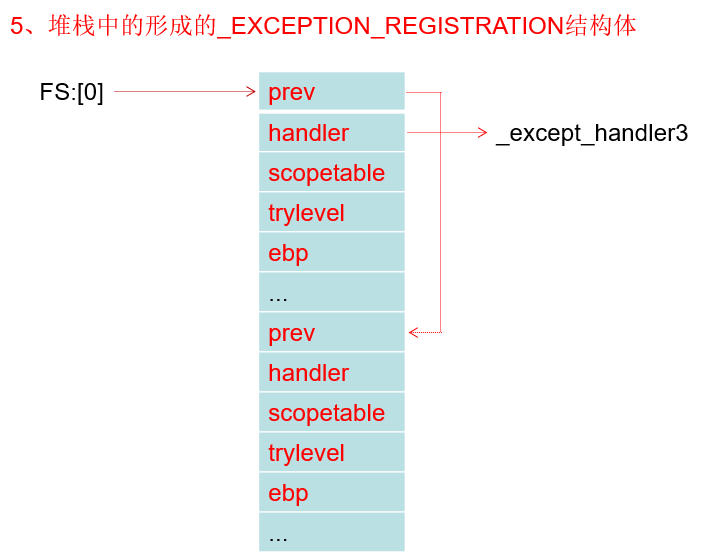
最后是通过context来恢复的，也就是最后将在3环修改后context的值放入到\_trap\_frame中，再进行恢复。

之后就是KiUserExceptionDispatcher函数









根据trylevel来找到发生异常的地方位于那个SEH中，相当于一个index，根据这个index来handle异常处理函数except\_handle函数来到scopetable中去索引，找到指定的项，例如index=2，那么就到scopetable中找到第三项，调用其中的IpfnFilter过滤函数，这个过滤函数的作用就是判断当前这个处理函数能否处理解决这个异常，如果不能就返回EXCEPTION\_CONTINUE\_SEARCH(0)，继续向下搜索，搜索的下一个目标就是scopetable中的previoustrylevel项（该项表明目前try处于哪个try中，也就是嵌套行的父try），直到当前previoustrylevel=-1表明已经到了最外层。如果过滤函数返回EXCEPTION\_EXECUTE\_HANDLER(1)，那么就执行scopetable中的异常处理函数，当异常处理函数处理完之后，就会直接跳转到try之后，直接跳出到try之外，不再执行try中代码。



如果VEH和SEH都不能处理解决这个异常，那么就要到最后一层的try，





当UnhandledExceptionFilter函数返回EXCEPTION\_EXECUTE\_HANDLER(1)，那么就执行except中的代码。

这里存在反调试，NtQueryInformationProcess函数查询Debugport来判断，是否被调试，如果被调试，那么就返回EXCEPTION\_CONTINUE\_SEARCH，之后就从RtlDispatchEXception函数返回，之后进行第二次分发异常，调用NtRaiseException函数。只有当没有被调试的时候才会调用注册的处理函数。

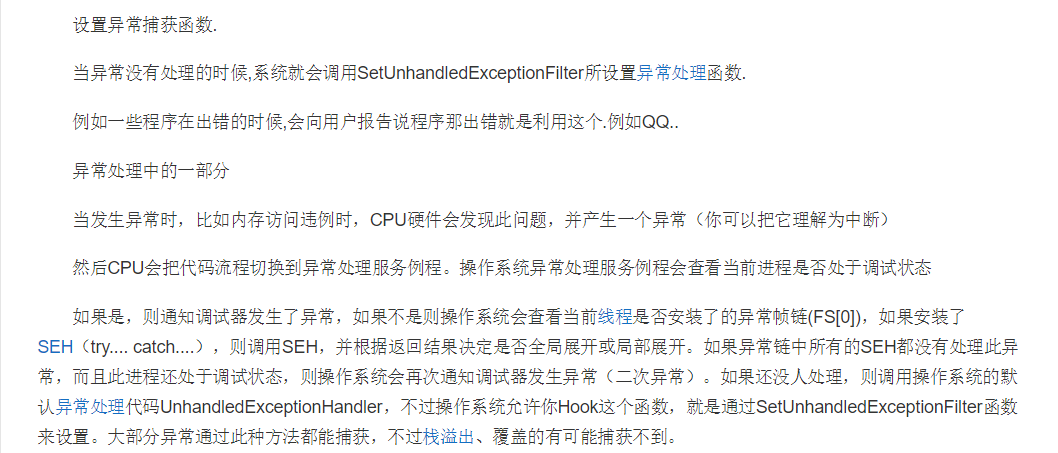


<https://blog.csdn.net/diamont/article/details/4259707>

讲的比较好

先说明一下：fs:[0]指向EXCEPTION\_REGISTRATION结构，也就是指向一个SEH结构

一定要弄清楚TIB，TEB，PEB



由于进程在创建所有的线程时都会安装顶层异常处理过程，并且把UEF（UnhandledExceptionFilter）作为异常过滤函数，所有的线程均会把UEF作为异常过滤函数，因此当调用UEF时，会检查用户是否设置了回调函数，如果设置了就会调用，可以通过SetUnhandledExceptionFilter来设置回调函数，



如上图压栈的时候是从下到上，先将结构体中下面的参数压栈。



